Terms used
segment and **processor** and **target** **address** and **mode** and **register** and **selector** and **stack** and **program** **counter**

Foun
28,22
134,83

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Improving prediction for procedure returns with return-address-stack repair mechanisms
Kevin Skadron, Pritpal S. Ahuja, Margaret Martonosi, Douglas W. Clark
November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.66 MB) Additional Information: full citation, references, citings, index terms

2 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 ARPS: a new real-time computer
Kenneth J. Thurber
October 1976 **ACM SIGARCH Computer Architecture News**, Volume 5 Issue 4
Full text available: pdf(1.14 MB) Additional Information: full citation, references

4 Performance effects of architectural complexity in the Intel 432
Robert P. Colwell, Edward F. Gehringer, E. Douglas Jensen
August 1988 **ACM Transactions on Computer Systems (TOCS)**, Volume 6 Issue 3
Full text available: pdf(3.45 MB) Additional Information: full citation, abstract, references, citings, index terms

The Intel 432 is noteworthy as an architecture incorporating a large amount of functionality that most other systems perform by software. It has, in effect, "migrated" this functionality from the software into the microcode and hardware. The benefits of functional migration have recently been a subject of intense controversy, with critics claiming that a complex architecture is inherently less efficient than a simple architecture with good software support. This paper examines t ...

5 Associative and Parallel Processors
Kenneth J. Thurber, Leon D. Wald

h c g e cf c

December 1975 **ACM Computing Surveys (CSUR)**, Volume 7 Issue 4

Full text available: pdf(2.62 MB) Additional Information: full citation, references, citings, index terms

6 DISE: a programmable macro engine for customizing applications

Marc L. Corliss, E. Christopher Lewis, Amir Roth

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(335.30 KB) Additional Information: full citation, abstract, references

Dynamic Instruction Stream Editing (DISE) is a cooperative software-hardware scheme for efficiently adding customization functionality---e.g, safety/security checking, profiling, dynamic code decompression, and dynamic optimization---to an application. In DISE, application customization functions (ACFs) are formulated as rules for macro-expanding certain instructions into parameterized instruction sequences. The processor executes the rules on the fetched instructions, feeding the executi ...

7 Data-Driven and Demand-Driven Computer Architecture

Philip C. Treleaven, David R. Brownbridge, Richard P. Hopkins

January 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 1

Full text available: pdf(4.14 MB) Additional Information: full citation, references, citings, index terms

8 An instruction fetch unit for a graph reduction machine

S. S. Thakkar, W. E. Hostmann

June 1986 **ACM SIGARCH Computer Architecture News , Proceedings of the 13th annual international symposium on Computer architecture**, Volume 14 Issue 2

Full text available: pdf(945.42 KB) Additional Information: full citation, abstract, references, index terms

The G-machine provides architecture support for the evaluation of functional programming languages by graph reduction. This paper describes an instruction fetch unit for such an architecture that provides a high throughput of instructions, low latency and adequate elasticity in the instruction pipeline. This performance is achieved by a hybrid instruction set and a decoupled RISC architecture. The hybrid instruction set consists of complex instructions that reflect the abstract architecture ...

9 A quantitative evaluation of interrupt handling capabilities in Ada

D. Strube

January 1989 **Proceedings of the conference on Tri-Ada '89: Ada technology in context: application, development, and deployment**

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, index terms

Projects considering the use of Ada for embedded weapon systems have performance as their primary concern. Despite the genuine interest throughout industry to commit to Ada for reasons of portability and maintainability, the language is still viewed with reservations when being considered for use in applications with high interrupt throughput or tasking requirements. There is increased demand from users for quantitative data on Ada's ability to handle deadlines in realtime. Such data can be ...

10 Design of a user-microprogrammable building block

Michael Kraley, Randall Rettberg, Philip Herman, Robert Bressler, Anthony Lake

November 1980 **Proceedings of the 13th annual workshop on Microprogramming**

Full text available: pdf(956.02 KB) Additional Information: full citation, abstract, references, citings, index terms

A user-microprogrammable computer has been developed for use as a building block in general-purpose and dedicated computer systems. The architecture is designed to be easily microprogrammed and features a 32-bit, vertically oriented microinstruction. The processor has a 135-nanosecond cycle time, either 16- or 20-bit macro data paths, and 1024 hardware registers. A significant fraction of the processor bandwidth may be budgeted for I/O processing to allow the substitution of microcode for e ...

11 A parallel embedded-processor architecture for ATM reassembly

Richard F. Hobson, P. S. Wong

February 1999 **IEEE/ACM Transactions on Networking (TON)**, Volume 7 Issue 1

Full text available: pdf(331.21 KB) Additional Information: full citation, references, citings, index terms

Keywords: ATM, embedded systems, medium access control, segmentation and reassembly

12 Virtual machine monitors: Xen and the art of virtualization

Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principles**

Full text available: pdf(168.76 KB) Additional Information: full citation, abstract, references, index terms

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.This paper presents Xen, an x86 virtual machine monit ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

13 The design and development of a dynamic program behavior measurement tool for the Intel 8086/88

R. J. Schwartz

June 1989 **ACM SIGARCH Computer Architecture News**, Volume 17 Issue 4

Full text available: pdf(906.01 KB) Additional Information: full citation, abstract, index terms

When modeling a computer system, it is necessary to study the system's dynamic behavior. Examples of this behavior are branching frequency and operating system usage patterns. The dynamic properties of the system characterize its performance. Models make assumptions about such behavior, but require genuine data to validate the assumptions. In this article, we present a measurement tool that will collect and analyze dynamic program information for the Intel 8086/88. The design and development of ...

14 Processor microarchitecture I: Partitioned first-level cache design for clustered microarchitectures

Paul Racunas, Yale N. Patt

June 2003 **Proceedings of the 17th annual international conference on Supercomputing**

Full text available: pdf(191.74 KB) Additional Information: full citation, abstract, references, index terms

The high clock frequencies of modern superscalar processors make the wire delay incurred in moving data across the processor chip a significant concern. As frequencies continue to increase, it will become more difficult for a centralized first level data cache to supply the timely data bandwidth required by superscalar processors.This paper presents a complete solution for the partitioning of the first level of the memory hierarchy. The first level data cache is split into several independent pa ...

Keywords: clustered microarchitecture, partitioned cache

15 Run-time checking in Lisp by integrating memory addressing and range checking

M. Sato, S. Ichikawa, E. Goto

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture**, Volume 17 Issue 3

Full text available: pdf(955.89 KB) Additional Information: full citation, abstract, references, index terms

This paper describes the BL addressing mode and the address tag in FLATS2 machine, which is a general-purpose MIMD computer now under construction. The BL addressing mode integrates memory accessing and range checking by hardware. Address tag is a bit in word, which indicates the capability

for memory access. Combining them together, efficient memory protection is provided at run-time. It reduces the cost of run-time type checking in Lisp by checking the address tag and the address of a poi ...

16 Streamlining data cache access with fast address calculation

Todd M. Austin, Dionisios N. Pnevmatikatos, Gurindar S. Sohi

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.58 MB) Additional Information: full citation, abstract, references, citings, index terms

For many programs, especially integer codes, untolerated load instruction latencies account for a significant portion of total execution time. In this paper, we present the design and evaluation of a fast address generation mechanism capable of eliminating the delays caused by effective address calculation for many loads and stores.Our approach works by predicting early in the pipeline (part of) the effective address of a memory access and using this predicted address to speculatively access the ...

17 Pipeline Architecture

C. V. Ramamoorthy, H. F. Li

January 1977 **ACM Computing Surveys (CSUR)**, Volume 9 Issue 1

Full text available: pdf(3.53 MB) Additional Information: full citation, references, citings, index terms

18 Architectural features of CASSM: A Context Addressed Segment Sequential Memory

G. J. Lipovski

April 1978 **Proceedings of the 5th annual symposium on Computer architecture**

Full text available: pdf(906.06 KB) Additional Information: full citation, abstract, references, citings, index terms

A Context Addressed Segment Sequential Memory (CASSM) was built to evaluate several techniques for nonnumeric processing. Herein, it is described from the architectural point-of-view. The basic architecture is introduced, information, data and storage structures are discussed, and the principles of disc searching used by CASSM are discussed.

19 Three simulator tools for teaching computer architecture: Little Man computer, and RTLSim

Cecile Yehezkel, William Yurcik, Murray Pearson, Dean Armstrong

December 2001 **Journal on Educational Resources in Computing (JERIC)**, Volume 1 Issue 4

Full text available: pdf(802.85 KB) Additional Information: full citation, abstract, citings, index terms

Teaching computer architecture (at any level) is not an easy task. To enhance learning, a critical mass of educators has begun using simulator visualizations of different computer architectures. Here we present three representative computer architecture simulators for learning which show that there is a growing consensus for computer simulation as a teaching tool for complex dynamic processes, such as underlying computer operations. Simulators also show the wide spectrum of pedagogical g ...

Keywords: Computer architecture simulators, education

20 Multithreading I: Master/slave speculative parallelization

Craig Zilles, Gurindar Sohi

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.31 MB) Publisher Site Additional Information: full citation, abstract, references, index terms

Master/Slave Speculative Parallelization (MSSP) is an execution paradigm for improving the execution rate of sequential programs by parallelizing them speculatively for execution on a multiprocessor. In MSSP, one processor---the master---executes an approximate version of the program to compute selected values that the full program's execution is expected to compute. The master's results are checked by slave processors that execute the original program. This validation is parallelized by cutting ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

h c g e cf c